Remarks

The above Amendments and these Remarks are in reply to the Office Action mailed February 10, 2006.

I. Summary of Examiner's Rejections

Prior to the Office Action mailed February 10, 2006, Claims 1-17 were pending in the Application. In the Office Action, all of the claims were rejected under 35 U.S.C. 103(a) as being unpatentable over Gall (U.S. Patent No. 6,480,862) in view of Sutton, ("Generalization in Reinforcement Learning: Successful Examples Using Sparse Coarse Coding", Advances in Neural Information Processing Systems 8, 1996, pp 1038-1044). The Specification was also objected to for various informalities.

II. Summary of Applicant's Amendment

The present Response amends the Specification; cancels Claims 1-6, 11, 13-17; amends Claims 7 and 12; and adds new Claims 18-29, leaving for the Examiner's present consideration Claims 7-10, 12, 14-16 and 18-29. Reconsideration of the Application is respectfully requested.

III. Amendments to the Specification

The present Response amends the Specification as shown above. Applicant respectfully submits that the proposed amendments correct informalities in the Specification, and that no new matter is being added.

IV. Claim Rejections under 35 U.S.C. §103(a)

In the Office Action mailed February 10, 2006, Claims 1-17 were rejected under 35 U.S.C. 103(a) as being unpatentable over Gall (U.S. Patent No. 6,480,862) in view of Sutton ("Generalization in Reinforcement Learning: Successful Examples Using Sparse Coarse Coding").

Claims 1-6, 11, 13 and 17

Claim 1-6, 11, 13 and 17 have been canceled, rendering moot the rejection of these claims. Applicant respectfully reserves the right to prosecute any originally presented or canceled claims in a continuing or future application.

Claim 7

Claim 7 has been amended to more clearly define the embodiment therein. As amended, Claim 7 defines:

7. (Currently Amended) A system for memory management comprising:
a computer system including a virtual machine operating thereon;
a memory space within said computer system and accessible by the virtual machine for the runtime storage and execution of applications; and,
a garbage collector that uses a reinforcement learning process to control the allocation of memory to applications within the memory space;
wherein the garbage collector performs the steps of
measuring system-wide and application-specific parameters to determine a current state of the memory space,
monitoring system events that contribute to the state,
performing actions by the garbage collector to adjust the allocation of memory,
calculating a reward, and measuring the system-wide and application-specific parameters to determine a new state associated with the reward, and,
repeating the steps as necessary to control the allocation of memory to applications within the memory space.

Claim 7, as currently amended, defines the garbage collector as performing the steps of measuring system-wide and application-specific parameters to determine a current state of the memory space, monitoring system events that contribute to the state, performing actions by the garbage collector to adjust the allocation of memory, calculating a reward, and measuring the system-wide and application-specific parameters to determine a new state associated with the reward. The steps are repeated as necessary to control the allocation of memory to applications within the memory space. Applicant respectfully submits that these features are not disclosed by the cited references.

The advantages of the embodiment currently defined by Claim 7 include that it allows the system to measure *system-wide and application-specific parameters*, such as the amount of allocated memory per time unit, or the last time a decision to garbage collect was made, and to use this information to determine a state of the system. The system is then allowed to performing *actions by the garbage collector* to adjust the allocation of memory. These action may include the decision to garbage collect or not, or to increase or reduce the heap size. A *new state* is then

created, together with a *reward*. The system uses this information to continually improve the memory-handling and other performance attributes of the virtual machine until a particular application terminates, or for a continuous or unspecified period of time.

Gall discloses a system and method for relation-based ordering of objects in an object heap. As described therein, data objects in an object heap are organized based upon access relationships between the data objects. Data objects that are accessed in close succession with one another are more likely to be located within the same page, and possibly within the same cache line, as one another. Consequently, when accessing such objects, the frequency of memory swapping within a multi-level memory architecture is often reduced. An access relationship may be based at least in part upon a likely temporal proximity of the accesses to such data objects, e.g., when a group of objects are typically accessed one after another during execution of a computer program. (Column 3, lines 5-22). In one embodiment, the access information for certain objects is maintained within an access order tree data structure. A root node represents the starting point for the tree under which all objects are referenced by a particular method. Some of the nodes can be referenced by other nodes, representing a chain of referenced objects. (Column 9, lines 29-49).

Sutton discloses a generalized reinforcement learning technique that uses sparse coarse coding. In particular, Sutton describes a variation of a SARSA algorithm that is applied to state-action pairs instead of states, and where the predictions are used as the basis for selecting actions. Sutton then applies the SARSA algorithm to tasks with a continuous state space, by combining it with a sparse, coarse-code function approximator known as the CMAC. (Sutton, Page 2).

It appears from the above description that Gall discloses a virtual machine and garbage collector architecture which benefits from a unique *ordering of the objects in the object heap*. The technique disclosed therein recognizes that during the typical execution of a particular computer program some objects are likely to be accessed one-after-another. This information can be stored in an order tree, with nodes in the tree representing which objects are referenced by a particular method. The information in the tree is then used to optimize the location of related nodes and hence objects in memory, which reduces memory-swapping, and provides benefits during garbage collection. However, Applicant respectfully submits that Gall does not appear to describe any modifications to the garbage collector itself. Indeed, Gall appears to disclose that various types of garbage collectors may be used consistent with the invention. (Column 6, lines 36-38). As such, Applicant respectfully submits that Gall does not teach a garbage collector that uses a

reinforcement learning process (or any modified process) to control the allocation of memory to applications within the memory space, as defined by Claim 7

Sutton discloses a *generalized reinforcement learning* technique. However, Applicant respectfully submits that Sutton does not disclose that this techniques can be applied to the particular problem of computer heap management. In particular, Applicant respectfully submits that Sutton does not disclose the reinforcement learning features of Claim 7, namely measuring system-wide and application-specific parameters to determine a current state of the memory space, monitoring system events that contribute to the state, performing actions by the garbage collector to adjust the allocation of memory, calculating a reward, and measuring the system-wide and application-specific parameters to determine a new state associated with the reward, and, repeating the steps as necessary to control the allocation of memory to applications within the memory space.

Furthermore, Applicant respectfully submits that, notwithstanding the comments provided above, it would not have been obvious to one of skill in the art to combine each of the cited references in the manner suggested, so as to anticipate the claimed embodiment. To establish a prima facie case of obviousness, three criteria must be met: (1) a suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine the reference teachings; (2) a reasonable expectation of success; and (3) the prior art references when combined must teach or suggest all the claim limitations. MPEP §2143. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, not in the applicant's disclosure. In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). In addition, the fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination. In re Mills, 916 F.2d 680, 16 USPQ2d 1430 (Fed. Cir. 1990).

In the present instance, Applicant has not been able to find any suggestion in the cited references themselves or in the knowledge generally available to one of ordinary skill in the art, that would render the resultant combination obvious or desirable. In particular, it appears the parties agree that Gall does not explicitly disclose or suggest the use of reinforcement learning. However, nor does Sutton appear to disclose techniques for memory management within a virtual machine.

In the Office Action it was submitted that it would have been obvious to incorporate the teachings of Sutton into those of Gall, because both Sutton and Gall are directed to temporal and

dynamic programming, and, more specifically Gall is directed to heap managed garbage collection in multiple computer programs executing simultaneously on the same computer system, while Sutton is directed to improving approximation, and estimation based on reinforcement learning and specifically implementing algorithm models that execute the task dynamics and applied dynamic programming backups. However, Applicant has been unable to find any suggestion or desire in Gall to modify the garbage collector described therein. Indeed, Gall appears to be designed to work with any generic garbage collector, and the techniques described therein are geared more toward a relational storing of objects in memory rather than any modified form of garbage collecting. Similarly, Applicant has been unable to find any suggestion or desire in Sutton to apply the techniques therein to memory management, and specifically management of the heap in a virtual machine. The use of reinforcement learning to manage memory in a virtual machine is indeed a useful feature of the present invention, but does not appear to be suggested by any of the cited references. As such, Applicant respectfully submits that the suggested combination of references lacks evidentiary support by the prior art.

In view of the comments provided above, Applicant respectfully submits that the embodiment defined by Claim 7 is neither anticipated by, nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

Claim 12

The comments provided above with respect to Claim 7 are hereby incorporated by reference. Claim 12 has been similarly amended to more clearly define the embodiment therein. For similar reasons as provided above with respect to Claim 7, Applicant respectfully submits that Claim 12, as amended, is likewise neither anticipated by, nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

Claims 8-10 and 14-16

Claims 8-10 and 14-16 depend from and include all of the features of Claims 7 and 12 respectively. Claims 8-10 and 14-16 are not addressed separately but it is respectfully submitted that these claims are allowable as depending from an allowable independent claim, and further in view of the comments provided above. Reconsideration thereof is respectfully requested.

V. Additional Amendments

Claims 18-29 have been newly added by the present Response. Applicant respectfully requests that new Claims 18-29 be included in the Application and considered therewith.

VI. Conclusion

In view of the above amendments and remarks, it is respectfully submitted that all of the Claims now pending in the subject patent application should be allowable, and reconsideration thereof is respectfully requested. The Examiner is respectfully requested to telephone the undersigned if he can assist in any way in expediting issuance of a patent.

Enclosed is a PETITION FOR EXTENSION OF TIME UNDER 37 C.F.R. §1.136 for extending the time to respond up to and including July 10, 2006.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 06-1325 for any matter in connection with this response, including any fee for extension of time, which may be required.

Respectfully submitted,

Date: July 10, 2006

By: _____
Karl Kenna
Reg. No. 45,445

Customer No.: 23910
FLIESLER MEYER LLP
Four Embarcadero Center, Fourth Floor
San Francisco, California 94111-4156
Telephone: (415) 362-3800